

TESTING PROCESS APPROACH & METHODOLOGY



Key Management Group

Fostering Partnerships



Proprietary Notice

All trademarks, services marks, trade names, logos, icons and other intellectual property rights (the "Intellectual Property") are proprietary to KMGI or used by KMGI as permitted through contractual arrangements with respective proprietors. The content of this testing process shall at all times continue to be the property of KMGI/ proprietors and will be strictly protected by copyright and other applicable laws. Any usage, reproduction, retransmission, distribution, dissemination, sale, publication or circulation of the said content without the express prior written consent of KMGI is strictly prohibited. Nothing in this document shall be construed as granting by implication or otherwise a license or a right to use such Intellectual Property, or indulge in reproduction or decompiling or sublicensing such Intellectual Property.

The users/clients expressly recognizes, admits and confirms that all the IPR rights in the development of this testing process is the sole property of KMGI and the user /client shall not claim or in any way assist, aid, permit or abet others in claiming any right title or interest there in or thereto. If required by KMGI the user/client shall cooperate in having KMGI register the IPR in this documentation, if deemed necessary.

© Unless otherwise acknowledged, the entire IPR (Intellectual Property Rights) in this testing process and contents thereof absolutely vests in KMGI. All rights reserved.

Contents

Contents

1. OVERVIEW.....	4
1.1. About KMG Testing Practice	4
1.2. Scope of this Document.....	4
1.3. About KMG.....	4
2. TESTING APPROACH FOR DEVELOPMENT PROJECT	5
2.1. Test Initiation Phase:	6
2.2. Test Planning Phase	8
2.3. Test Execution Phase	10
2.4. Test Reporting & Analysis Phase	11
3. TESTING APPROACH FOR MAINTENANCE PROJECT	13
3.1. Document High Level Flow of the Application	13
3.2. Create Unit Test Cases of Tickets/Enhancements.....	13
3.3. Prioritizing the Modules	13
3.4. Functional Test Pack	13
3.5. Development of Regression Test Cases	14
3.6. Tickets/Enhancement Testing	14
3.7. Regression Testing	14
4. SUMMARY.....	16

1. OVERVIEW

1.1. ABOUT KMG TESTING PRACTICE

KMG has an established testing practice and provides services covering the entire span of testing domain. These include developing Test Strategy, Test Plan, Developing functional/regression test cases & scripts as well as executing the scripts manually or using test automation tool.

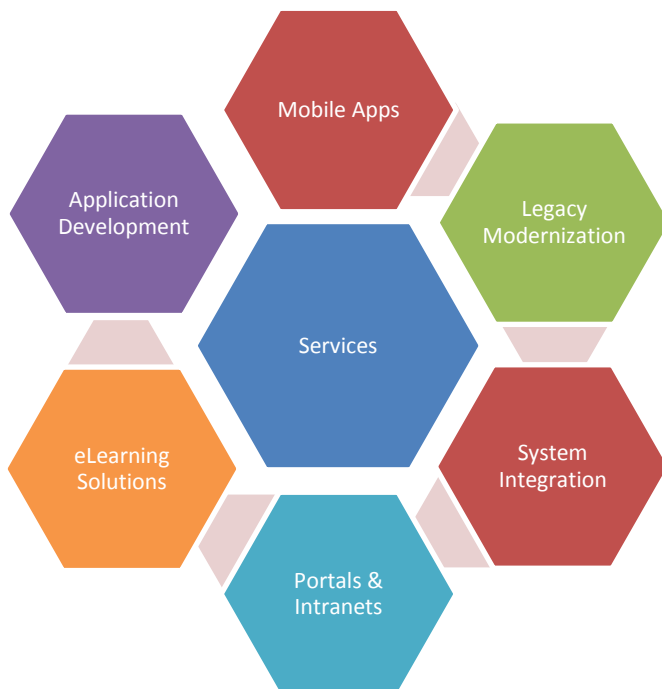
KMG has an established centralized, independent testing department - a Testing Centre of Excellence within the organization. Testing COE primarily focus on processes, people, tools, domain knowledge and best practices, with an emphasis on continual process improvements.

1.2. SCOPE OF THIS DOCUMENT

The objective of this document is to showcase KMG's testing approach and practice for all Development and Maintenance projects in KMG. This document details out when code freeze is done and how testing is carried out through its different phases within agreed upon timelines.

1.3. ABOUT KMG

KMG is a global software development, application maintenance and BPO operations company, which provides premium IT solutions worldwide using Microsoft, IBM & Java Technologies. In a wide-open sea of countless software development companies, KMG distinguishes itself as a company driven by excellence.



KMG was established in 1990 and is among Top 10 fastest growing Indian-owned companies in the US. It is also rated among top 50 software companies in India.

KMG has its headquarters in New York with multiple Offshore Development Centers in India. Our onsite-offshore model and industry expertise has enabled the company to enter into long-term, mutually beneficial strategic partnerships with many Fortune 500/1,000 companies. Unlike most other Indian-based software services firms, KMG maintains a large development team in the US. This team is used to interact with the client & provide a longer overlap to the users.

With a strength of approx. 60 professionals in the US supported by another 350 in India, KMG has resource expertise that covers Microsoft/.NET technologies, Java/J2EE, Mainframe, IBM iSeries (COBOL/RPG), Software Testing, BI, Open Source and other varied technologies.

KMG provides software maintenance, development and testing, and BPO solutions to large and medium sized insurance, banking, financial service, healthcare and government organizations throughout the world. Indeed, the company has aggressively captured a niche market in the property and casualty insurance sector in the USA.

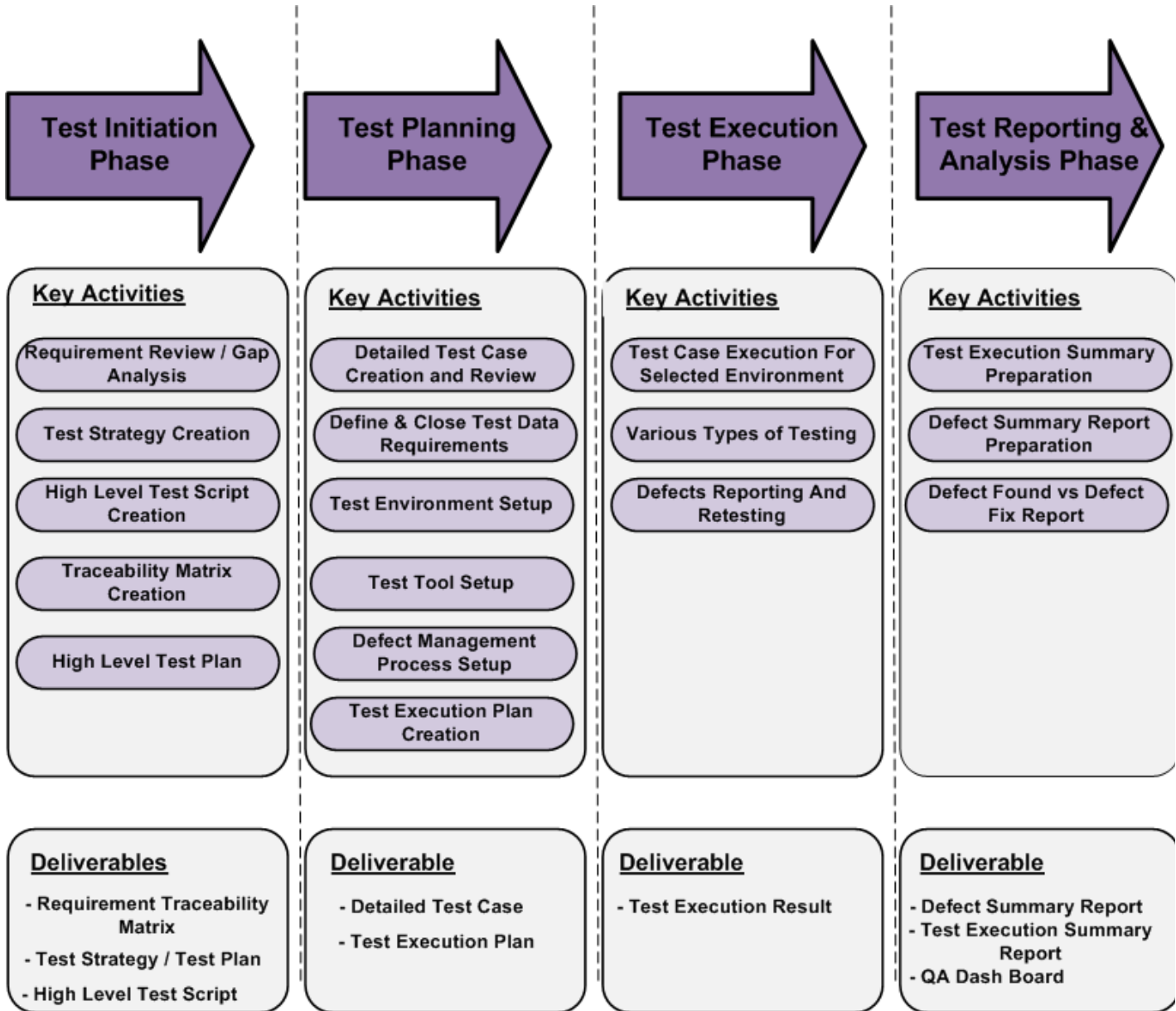
KMG has a 600-seat state-of-art development center near Chandigarh (North West part of India). This center houses a large training center for providing training to in-house / external resources on various technologies (including legacy systems). The center also houses the teams for QA testing & BPO processes.

2. TESTING APPROACH FOR DEVELOPMENT PROJECT

There are mainly four phases of testing for a development project:

- Test Initiation Phase
- Test Planning Phase
- Test Execution Phase
- Test Reporting & Analysis Phase

Chart below depicts these four phases:



KMG QA Process For Development Project

2.1. TEST INITIATION PHASE

Following testing activities are carried out during test initiation phase:

2.1.1 REQUIREMENT REVIEW/GAP ANALYSIS

The requirement specification document, Use cases and other project planning documents are considered as inputs for this activity. A walkthrough session with business analysis team is arranged for reviewing requirements document(s) and all issues regarding requirements are discussed and closed.

2.1.2 TEST STRATEGY CREATION

Test strategy is a document detailing the overall testing approach to be followed during entire testing life cycle of the project. Following are the major contents of test strategy:

- Purpose of the document
- Overview of the Application
- Objective of Testing
- Traceability Requirements
- Scope of Testing
- Test Approach
- Different type of testing to be performed
- Entry and Exit Criteria
- Test Suspension and Resumption Criteria
- Script Pass/Fail Criteria
- Test Environment Setup- Hardware, Operating System etc.
- Test Data Setup Process
- Defect Management Process
- Testing Tool to be used
- Communication Plan
- Roles and Responsibilities
- Assumption and Dependencies
- Risks
- Test Deliverables

Test strategy is prepared by KMG Test Manager and shared with client for review and approval.

2.1.3 HIGH LEVEL TEST SCRIPT CREATION

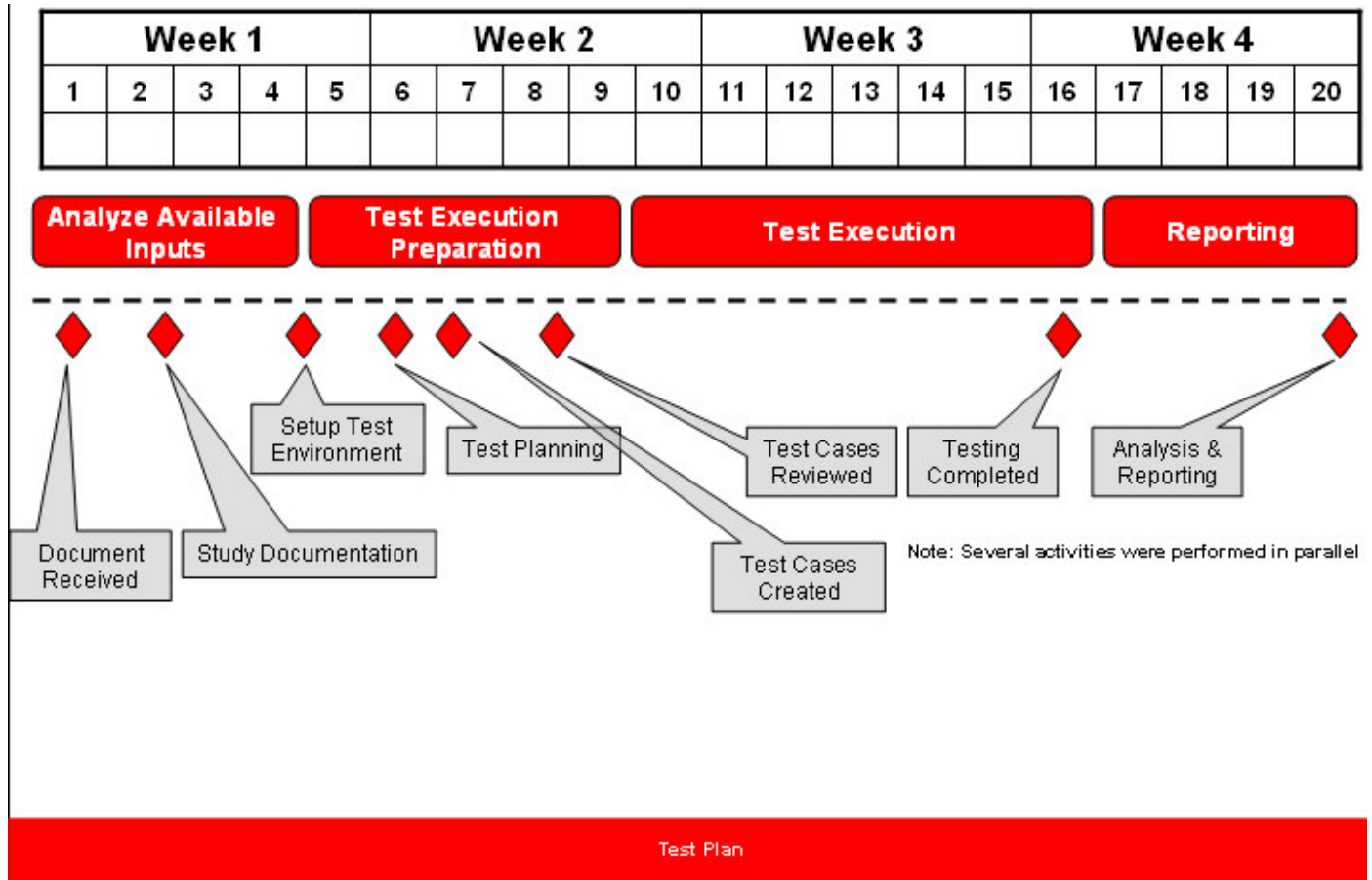
High Level test script is nothing but the test case objective. Test case objective is the description of what would be tested to cover the functionality of the application. The High Level Description should be self-sufficient to explain to the reader what is being tested.

2.1.4 TRACEABILITY MATRIX

Traceability matrix defines the mapping the customer's requirements and the prepared test cases. It helps in determining complete coverage of requirement specification. Tracing requirements from development to testing establishes the confidence that test cases are covering the developed functional requirements.

2.1.5 HIGH LEVEL TEST PLAN

Test Plan provides an overview of the major milestones of testing along with the target dates. Below snapshot is a sample high level plan that depicts various testing activities carried out during entire testing life cycle.



Test Initiation Phase Deliverables

Following are major deliverables of this phase and will serve as prerequisite for next phase:

- Traceability Matrix
- High Level Test Scripts
- High Level Test Plan

2.2. TEST PLANNING PHASE

Following testing activities are carried out during test planning phase:

2.2.1 DETAILS TEST CASE CREATION AND REVIEW

Detailed test cases contain low level information of test cases like Application Name, Release Name, Tester Name, Creation date, Module, Function, Steps to Test, Expected Result, and Actual Result etc.

These test cases are reviewed by internal testing team and then by client through test case review process.

2.2.2 TEST DATA REQUIREMENTS

Once test case development is finished, test data creation process is set up. Testing team may create dummy data in the application and that data can be used to perform testing. Data requirements are also given to the client incase client wants testing to be performed on specific test data.

There might be a possibility that client is already using an old application and wants KMG to perform testing on existing data. In that case, either data migration process is set up or a data dump is provided to KMG's testing team.

2.2.3 TEST ENVIRONMENT SETUP

All testing activities are managed across following four basic environments:

- Development Environment
- Test Environment
- Staging/Pre-Production Environment
- Production Environment

Infrastructure team helps in setting up these test environments.

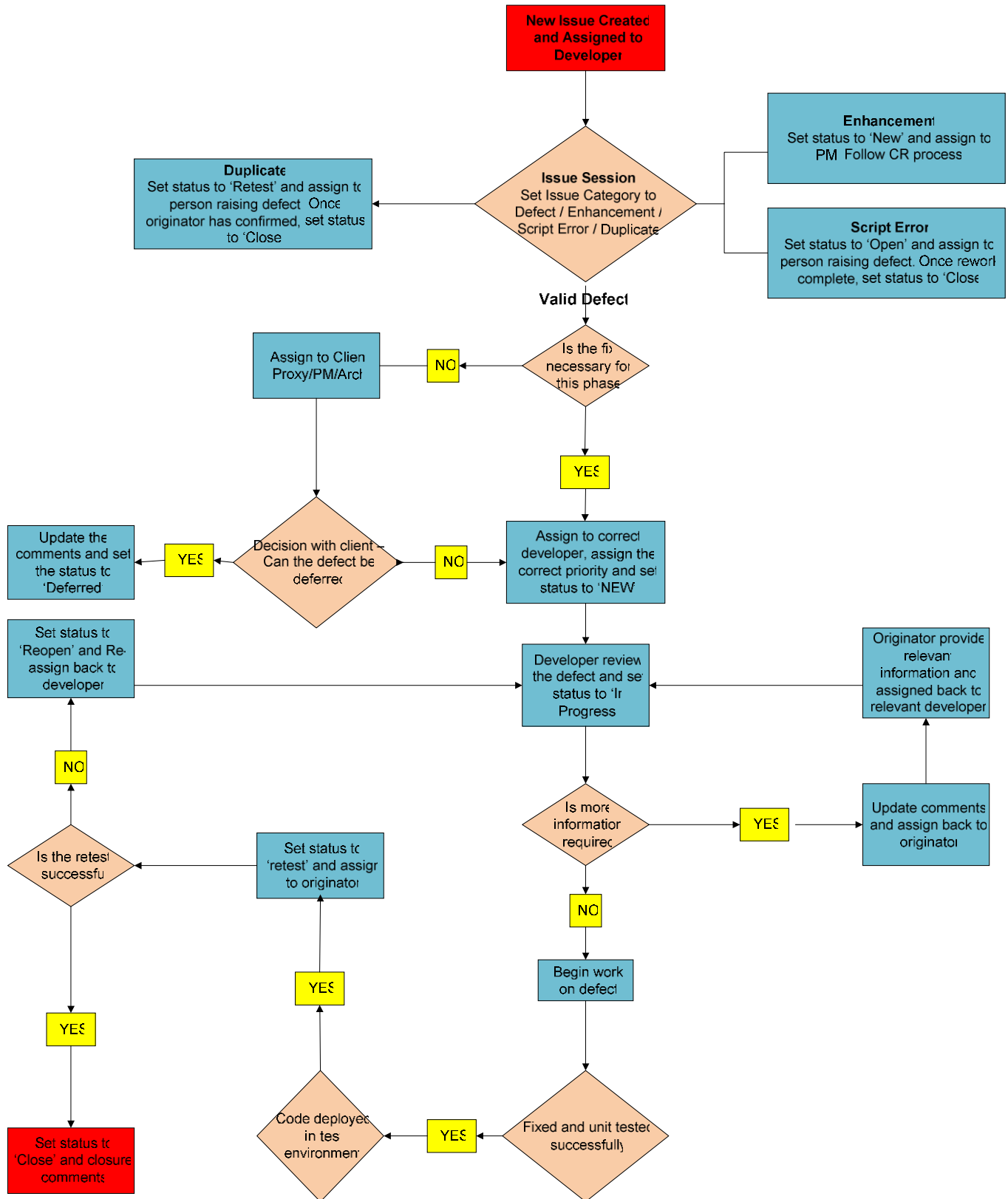
2.2.4 TEST TOOL SETUP

Various different type of testing tools are set up in test lab machines depending on the project's requirement. Testing tool may be a Test automation tool, Performance Testing tool, Test Management tool etc.

2.2.5 DEFECT MANAGEMENT PROCESS SET UP

All the defects found in the testing shall be logged in the defect management tool. Tester shall provide all necessary information required to replicate the defect to the developer. All defects will go through a defined life cycle as shown in below sample diagram:

Sample Defect Management Process



2.2.6 TEST EXECUTION PLAN CREATION

Test execution plan is created either by Test Manager or Test lead for each planned build. This plan details out when different type of testing to be performed for a particular test execution cycle.

Test Planning Phase Deliverables

Following are major deliverables of this phase and will serve as prerequisite for next phase:

- Detailed Test Cases
- Test Execution Plan

2.3. TEST EXECUTION PHASE

Following testing activities are carried out during test planning phase:

2.3.1 TEST CASE EXECUTION FOR SELECTED ENVIRONMENT

All identified test cases are executed as per the test execution plan for selected environment. At first testing by QA team is basically performed in QA environment which is isolated from development environment, then testing is performed in UAT environment by UAT team and then in staging or Pre-Production environment.

2.3.2 VARIOUS TYPES OF TESTING

Following various types of functional and non-functional testing are performed depending upon the project need during test execution phase:

Functional Testing

- Sanity Testing
- Integration Testing
- System Testing
- GUI Testing
- Regression Testing
- End to End Testing
- Exploratory Testing

Non Functional Testing

- Browser Testing
- Web Accessibility Testing
- Performance Testing
- Load Testing
- Security Testing
- Failover and Disaster Recovery Testing

2.3.3 DEFECTS REPORTING AND RETESTING

All defects encountered during testing are reported to concerned manager. A defect reporting follows a selected defect management process. Testing team is notified after defects are fixed and then a defect re-testing is performed to make sure if reported defects are fixed.

Test Execution Phase Deliverables

Following are major deliverables of this phase and will serve as prerequisite for next phase:

- Test Execution Result

2.4. TEST REPORTING & ANALYSIS PHASE

Following testing activities are carried out during test planning phase:

2.4.1 DEFECT SUMMARY REPORT PREPARATION

Once the test execution for selected execution cycle is over, a defect summary report is prepared containing all defects encountered along with the severity and functional area of the application.

2.4.2 TEST EXECUTION SUMMARY REPORT PREPARATION

Test execution summary contains status of all defects encountered for each execution cycle. This report helps in determining stability of particular functionality across each test execution cycle.

2.4.3 DEFECT FOUND VS DEFECT FIX REPORT PREPARATION

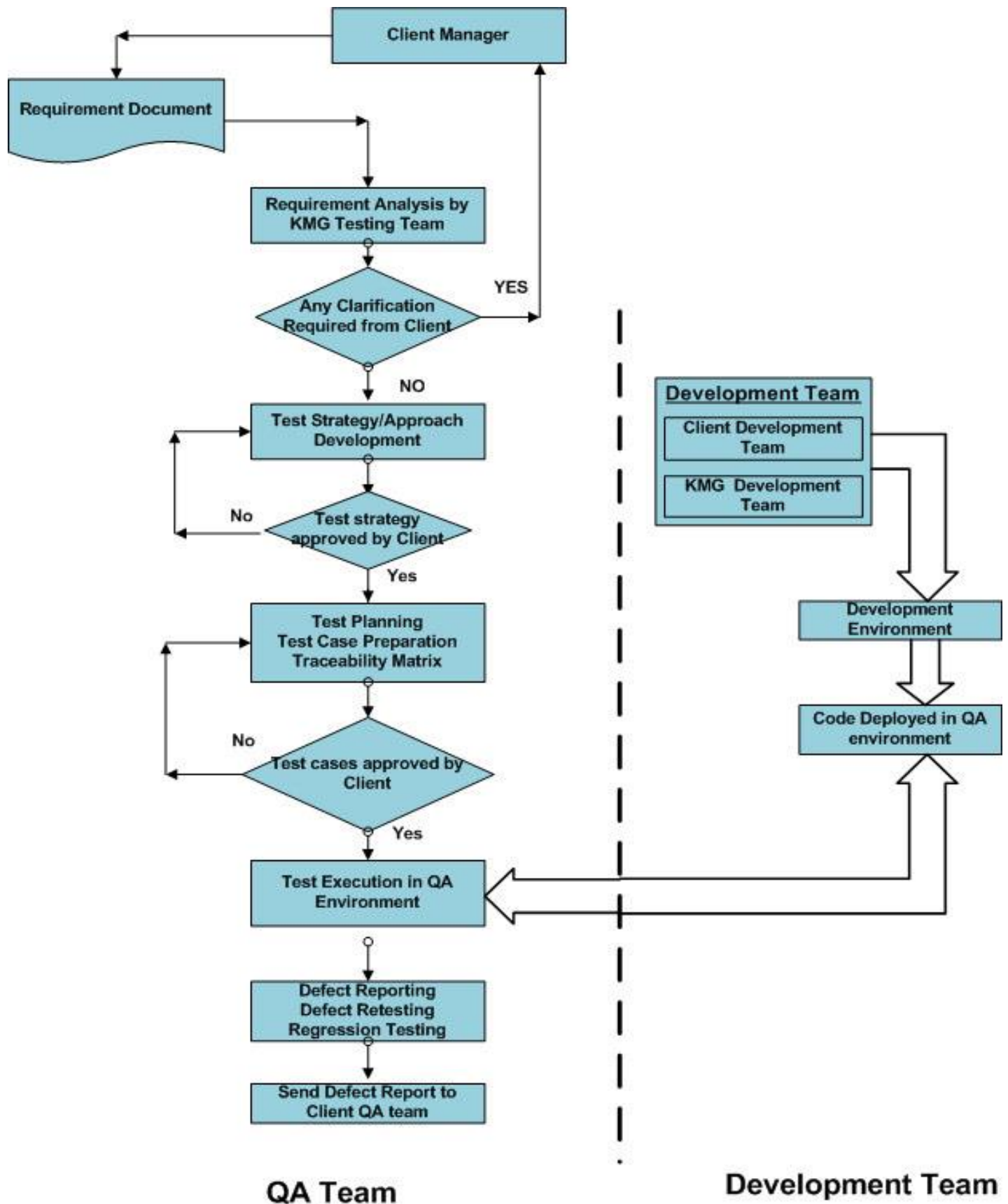
Defect found vs Defect fix report enables project manager to measure how many defects are fixed everyday against defects reported every day.

Test Execution Phase Deliverables

Following are major deliverables of this phase and will serve as prerequisite for next phase:

- Defect Summary Report
- Test Execution Summary Report
- QA Dash board

Below diagram depicts KMG QA Process Definition:



3. TESTING APPROACH FOR MAINTENANCE PROJECT

This section explains the overall testing approach that is followed for testing enhancements and developing the functional/Regression test cases for an enhancement project.

3.1. DOCUMENT HIGH LEVEL FLOW OF THE APPLICATION

- Prepare and document high level flow diagrams identifying different modules of the enhancement/ticket being tested.
- Find out the interaction between the modules and critical areas of the application.

3.2. CREATE UNIT TEST CASES OF TICKETS/ENHANCEMENTS

The unit cases developed based on the tickets received by the client’s support team would provide a part of the overall picture.

- Develop and document the unit test cases based on the problem request, change request and enhancement request
- This will be slow process & will depend on the number & type of tickets received
- It is proposed that this process will be a ongoing process and continue as long as new enhancements are provided

3.3. PRIORITIZING THE MODULES

The following guiding principles will be used while prioritizing the test cases

- Areas which are prone to heavy changes
- Risk to the business

A list of module with their priority is provided to Client for approval. Below is the sample template:

S.No	Module Name	Priority

3.4. FUNCTIONAL TEST PACK

Functional test pack will contain following test cases:

- Unit test cases of all subsequent tickets/enhancements covering new functionality of the application
- Existing functional test cases of the application.

3.5. DEVELOPMENT OF REGRESSION TEST CASES

Regression test cases will be the subset of functional test cases and will be derived from the functional test pack based on the priority of the business function and impacted areas identified after modification are done on existing application.

3.6. TICKETS/ENHANCEMENT TESTING

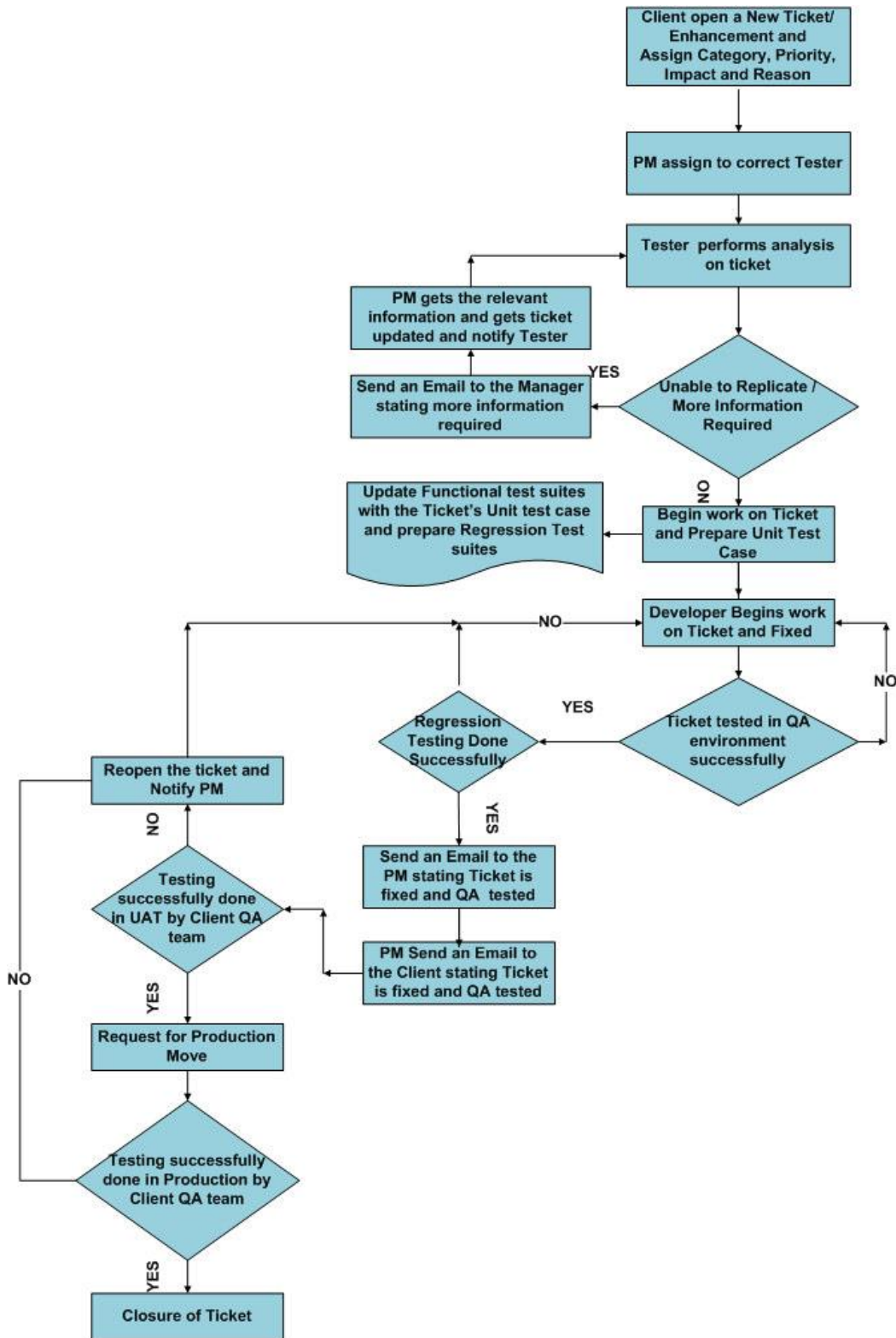
Once development and testing in Dev environment is finished, ticket is promoted to QA environment and then testing is performed by QA team. QA team not only executes the unit test cases of ticket but also execute the test cases of functionality that is impacted due to the enhancement done for the ticket.

3.7. REGRESSION TESTING

Regression test cases are executed to make sure that the existing functionality of the application has not been compromised by the modification to different parts of the application. Functional testing will be done for functionality newly added in the build, whereas the already existing functionality will be subjected to Regression testing.

Regression testing involves testers to running defined set of test cases on the system to ensure that no new bugs have been introduced in the system because of any recent modifications in the code. No separate test cases will be written for regression but functional test cases that retire from a build serve as regression test cases for subsequent build.

Below is the snapshot of high level Testing approach for Enhancement/Support project:



Phase – TEST INITIATION

Phase – TEST PLANNING

Phase – TEST EXECUTION

Phase – TEST REPORTING

KMG Testing Process For Enhancement/ Support Project

4. SUMMARY

Testing process is the planned and systematic set of activities that ensures software processes and products conform to requirements, standards, and procedures. Software development is a process full of risks and these risks are both technical and programmatic. The goal of testing process is to reduce these risks. For example, coding standards are established to ensure the delivery of quality code.

Even when standards are set, there is no explicit process of assuring that all codes meet the standards. Similarly, lack of an Error Management and Defect Life Cycle workflow increases the risk that problems in the software will be forgotten and not corrected.

The Testing Process is therefore an essential step in a software development life cycle to reduce these risks, and to assure quality in both the workflow and the final product.

This document presents a process for testing an application under development phase as well as maintenance phase. This also details out the planned and systematic set of various testing activities across the different phases of testing. Each of the four phases of Testing Lifecycle will incorporate QA activities and deliverables that reduce the risk of common testing problems.

The concepts that are outlined in the process apply regardless of the software development life cycle approach followed. Testing is sometime incorrectly thought as an after-the-fact activity; performed after programming is done. However, the testing process described in this document ensures that testing is involved throughout the SDLC in order to bring out a quality product.